

# Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date: 13/02/23

Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F
---------------------	-----------------------	------------

Week# \_\_\_\_4\_\_\_\_ Program Number: \_\_\_\_1\_\_

Title of the Program

**Write an ALP to add two 64 bit numbers loaded from memory and store the result in memory.**

I.ARM Assembly Code:

```
.data
a: .word 31258231, 67393412
b: .word 12350294, 23401830
c: .word 0,0

.text
LDR R0, =a
LDR R1, =b
LDR R2, =c
LDR R4, [R0]
LDR R5, [R1]
ADD R4, R4, R5
STR R4, [R2]
LDR R4, [R0], #4
LDR R5, [R1], #4
ADD R5, R4, R5
STR R5, [R2], #4
SWI 0x011
```

[illegible]

Date: 13/02/23

Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F
---------------------	-----------------------	------------

Week# \_\_\_\_4\_\_\_\_

Program Number: \_\_\_\_2\_\_\_\_

Title of the Program

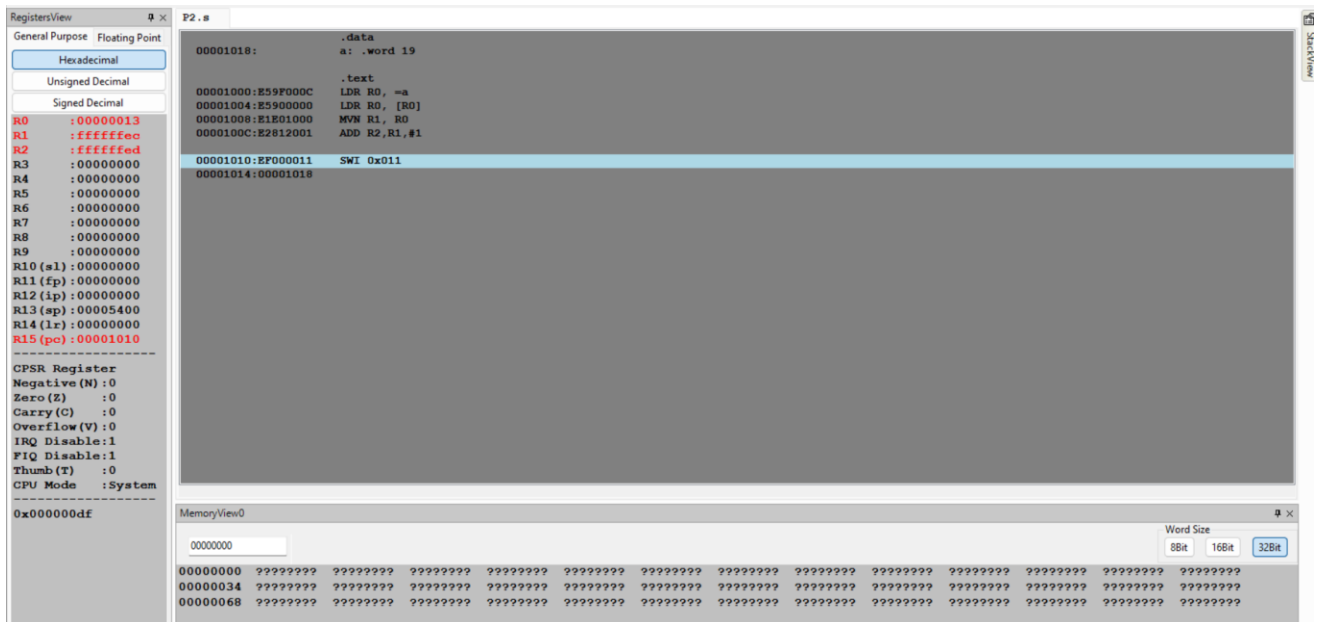
**Write an ALP to find 1's and 2's complement of a 32 bit number**

I.ARM Assembly Code:

```
.data
a: .word 19

.text
LDR R0, =a
LDR R0, [R0]
MVN R1, R0
ADD R2, R1, #1
SWI 0x011
```

II. Output Screen Shot (One)



Date: 13/02/23

Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F
---------------------	-----------------------	------------

Week# 4 Program Number: 3

# Title of the Program

**Write an ALP to scan a 32 bit number if it is negative or positive**

## I. ARM Code:

```
.data
a: .word 20
p: .ASCIIZ "positive"
n: .ASCIIZ "negative"

.text
LDR R1,=a
LDR R2,[R1]
CMP R2, #0
BLT min
LDR R0, =p
B END

min:
    LDR R0, =n

END:
    SWI 0x02
    SWI 0x11
.end
```

## II. Output Screen Shot (One) :

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 00001034  
R1 : 00001030  
R2 : 00000014  
R3 : 00000000  
R4 : 00000000  
R5 : 00000000  
R6 : 00000000  
R7 : 00000000  
R8 : 00000000  
R9 : 00000000  
R10 (s1) : 00000000  
R11 (fp) : 00000000  
R12 (ip) : 00000000  
R13 (sp) : 00011400  
R14 (lr) : 00000000  
R15 (pc) : 00001020  
  
CPSR Register  
Negative (N) : 0  
Zero (Z) : 0  
Carry (C) : 1  
Overflow (V) : 0  
IRQ Disable: 1  
FIQ Disable: 1  
Thumb (T) : 0  
CPU Mode : System  
  
0x200000df

CodeView

P3.o

```

.data
00001030:00000014 a: .word 20
00001034:69736F70 p: .ASCII "positive"
:65766974
:00
0000103D:6167656E n: .ASCII "negative"
:65766974
:00

.text
00001000:E59F101C LDR R1,=a
00001004:E5912000 LDR R2,[R1]
00001008:E3520000 CMP R2, #0
0000100C:BA000001 BLT min
00001010:E59F0010 LDR R0,=p
00001014:EA000000 B END

min:
00001018:E59F000C LDR R0,=n

END:
0000101C:EF000002 SWI 0x02
00001020:EF000011 SWI 0x11
00001024:00000000 .end...
:00000004
:0000000D

```

OutputView WatchView MemoryView0

Console

```

positive
Execution ending, Instruction Count:8 Elapsed Time:00:00:00.0060057
Instructions per second:1332
Execution starting ...
positive
Execution ending, Instruction Count:8 Elapsed Time:00:00:00.0071999
Instructions per second:1111

```

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0 : 0000103d  
R1 : 00001030  
R2 : ffffffff1  
R3 : 00000000  
R4 : 00000000  
R5 : 00000000  
R6 : 00000000  
R7 : 00000000  
R8 : 00000000  
R9 : 00000000  
R10 (s1) : 00000000  
R11 (fp) : 00000000  
R12 (ip) : 00000000  
R13 (sp) : 00011400  
R14 (lr) : 00000000  
R15 (pc) : 00001020  
  
CPSR Register  
Negative (N) : 1  
Zero (Z) : 0  
Carry (C) : 1  
Overflow (V) : 0  
IRQ Disable: 1  
FIQ Disable: 1  
Thumb (T) : 0  
CPU Mode : System  
  
0xa00000df

CodeView

P3.o

```

.data
00001030:FFFFFFF1 a: .word -15
00001034:69736F70 p: .ASCII "positive"
:65766974
:00
0000103D:6167656E n: .ASCII "negative"
:65766974
:00

.text
00001000:E59F101C LDR R1,=a
00001004:E5912000 LDR R2,[R1]
00001008:E3520000 CMP R2, #0
0000100C:BA000001 BLT min
00001010:E59F0010 LDR R0,=p
00001014:EA000000 B END

min:
00001018:E59F000C LDR R0,=n

END:
0000101C:EF000002 SWI 0x02
00001020:EF000011 SWI 0x11
00001024:00000000 .end...
:00000004
:0000000D

```

OutputView WatchView MemoryView0

Console

```

Loading assembly language file C:\Nikhil\school-work\Programs\4th Sem\MPCA\Week 4\P3.s
Execution starting ...
negative
Execution ending, Instruction Count:7 Elapsed Time:00:00:00.0047629
Instructions per second:1469

```

Date: 13/02/23

Name: Nikhil Girish

SRN:  
PES2UG21CS334

Section: F

Week# 4

Program Number: 4

Title of the Program

**Write an ALP to find the number of zeroes, positive and negative numbers in a given array**

I. ARM Code:

```
.data
arr: .word 0,3,-1,6,3,9,-4,-2,1,0

.text
LDR R0,=arr
MOV R1,#0
MOV R2,#0
MOV R3,#0
MOV R4,#0
MOV R5,#0

START:
    CMP R2,#10
    BEQ END
    LDR R1,[R0,R2,LSL#2]
    CMP R1,#0
    ADDEQ R3,R3,#1
    ADDGT R4,R4,#1
    ADDMI R5,R5,#1
    ADD R2,R2,#1
    B START
END:
    MOV R1,#0
    MOV R2,#0
    SWI 0x011
```

II. Output View:





Date:13/02/23

Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F
---------------------	-----------------------	------------

Week# \_\_\_\_4\_\_\_\_

Program Number: \_\_\_\_5\_\_

Title of the Program

**Write an ALP to count the number of 1's and 0's in a given 32 bit number.**

I. ARM Code:

```
.data
a: .word 11
ODD: .ASCIZ "Odd Parity"
EVEN: .ASCIZ "Even Parity"

.text
LDR R0,=a
LDR R0,[R0]
MOV R1,#0
MOV R2,#0
MOV R3,#0
MOV R4,#0
START:
    CMP R3,#32
    BEQ END
    TST R0,#1
    BEQ ZERO
    ADD R2,R2,#1
    B CONTINUE
ZERO:
    ADD R1,R1,#1
CONTINUE:
    MOV R0,R0,LSR#1
    ADD R3,R3,#1
```

```
    B START
END:
    TST R2, #1
    LDREQ R0, =EVEN
    LDRNE R0, =ODD
    MOV R3, #0
    MOV R4, #0
    SWI 0x02
    SWI 0x11
```

II. Output Screen Shot (One):

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0

: 0000106c

R1

: 0000001d

R2

: 00000003

R3

: 00000000

R4

: 00000000

R5

: 00000000

R6

: 00000000

R7

: 00000000

R8

: 00000000

R9

: 00000000

R10 (s1)

: 00000000

R11 (fp)

: 00000000

R12 (ip)

: 00000000

R13 (sp)

: 00011400

R14 (lr)

: 00000000

R15 (pc)

: 00001058

CPSR Register

Negative (N) : 0

Zero (Z) : 0

Carry (C) : 1

Overflow (V) : 0

IRQ Disable: 1

FIQ Disable: 1

Thumb (T) : 0

CPU Mode : System

0x200000df

CodeView

P5.o

00001000:E59F0054 LDR R0,=a

00001004:E5900000 LDR R0,[R0]

00001008:E3A01000 MOV R1,#0

0000100C:E3A02000 MOV R2,#0

00001010:E3A03000 MOV R3,#0

00001014:E3A04000 MOV R4,#0

START:

00001018:E3530020 CMP R3,#32

0000101C:0A000007 BEQ END

00001020:E3100001 TST R0,#1

00001024:0A000001 BEQ ZERO

00001028:E2822001 ADD R2,R2,#1

0000102C:EA000000 B CONTINUE

ZERO:

00001030:E2811001 ADD R1,R1,#1

CONTINUE:

00001034:E1A000A0 MOV R0,R0,LSR#1

00001038:E2833001 ADD R3,R3,#1

0000103C:EAF7FFF5 B START

END:

00001040:E3120001 TST R2,#1

00001044:059F0014 LDREQ R0,=EVEN

00001048:159F0014 LDRENE R0,=ODD

0000104C:E3A03000 MOV R3,#0

00001050:E3A04000 MOV R4,#0

00001054:EF000002 SWI 0x02

00001058:EF000011 SWI 0x11...

:00000000

:0000000F

:00000004

OutputView

WatchView

MemoryView0

Console

Loading assembly language file C:\Nikhil\school-work\Programs\4th Sem\MPCA\Week 4\P5.s

Execution starting ...

Odd Parity

Execution ending, Instruction Count:274 Elapsed Time:00:00:00.3559508

Instructions per second:769

RegistersView

General Purpose Floating Point

Hexadecimal

Unsigned Decimal

Signed Decimal

R0

: 00001077

R1

: 0000001e

R2

: 00000002

R3

: 00000000

R4

: 00000000

R5

: 00000000

R6

: 00000000

R7

: 00000000

R8

: 00000000

R9

: 00000000

R10 (s1)

: 00000000

R11 (fp)

: 00000000

R12 (ip)

: 00000000

R13 (sp)

: 00011400

R14 (lr)

: 00000000

R15 (pc)

: 00001058

CPSR Register

Negative (N) : 0

Zero (Z) : 1

Carry (C) : 1

Overflow (V) : 0

IRQ Disable: 1

FIQ Disable: 1

Thumb (T) : 0

CPU Mode : System

0x600000df

CodeView

P5.o

.data

a: .word 20

00001068:00000014

0000106C:2064644F ODD: .ASCIZ "Odd Parity"

:69726150

:007974

00001077:6E657645 EVEN: .ASCIZ "Even Parity"

:72615020

:00797469

.text

00001000:E59F0054 LDR R0,=a

00001004:E5900000 LDR R0,[R0]

00001008:E3A01000 MOV R1,#0

0000100C:E3A02000 MOV R2,#0

00001010:E3A03000 MOV R3,#0

00001014:E3A04000 MOV R4,#0

START:

00001018:E3530020 CMP R3,#32

0000101C:0A000007 BEQ END

00001020:E3100001 TST R0,#1

00001024:0A000001 BEQ ZERO

00001028:E2822001 ADD R2,R2,#1

0000102C:EA000000 B CONTINUE

ZERO:

00001030:E2811001 ADD R1,R1,#1

CONTINUE:

00001034:E1A000A0 MOV R0,R0,LSR#1

00001038:E2833001 ADD R3,R3,#1

0000103C:EAF7FFF5 B START

END:

OutputView

WatchView

MemoryView0

Console

Loading assembly language file C:\Nikhil\school-work\Programs\4th Sem\MPCA\Week 4\P5.s

Execution starting ...

Even Parity

Execution ending, Instruction Count:273 Elapsed Time:00:00:00.1398084

Instructions per second:1952

# Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date: 13/02/23

Name: Nikhil Girish	SRN: PES2UG21CS334	Section: F
---------------------	-----------------------	------------

Week# \_\_\_\_4\_\_\_\_

Program Number: \_\_\_\_6\_\_

Title of the Program

**Write an ALP to check the given number has odd or even number of 1's and display the result. (Even Parity and Odd Parity)**

I. ARM Code:

```
.data
a: .word 69
odd: .asciz "ODD PARITY"
even: .asciz "EVEN PARITY"

.text
LDR R0,=a
LDR R0,[R0]
MOV R1,#0
MOV R2,#0
MOV R3,#0
MOV R4,#0
START:
    cmp R3,#32
    BEQ END
    TST R0,#1
    BEQ CONTINUE
```

```

        ADD R2,R2,#1
CONTINUE:
        MOV R0,R0,LSR#1
        ADD R3,R3,#1
        B START
EVEN:
        LDR R0,=even
        SWI 0x02
        SWI 0x011

END:
        MOV R3,#0
        MOV R4,#0
        TST R2,#1
        BEQ EVEN
        LDR R0,=odd
        SWI 0x02
        SWI 0x011

```

## II. Output Screen Shot (One):

The screenshot displays a debugger interface with three main panes: RegistersView, CodeView, and OutputView.

**RegistersView:** Shows the state of 16 registers (R0-R15). R0 is 00001070, R1 is 00000000, R2 is 00000003, R3 is 00000000, R4 is 00000000, R5 is 00000000, R6 is 00000000, R7 is 00000000, R8 is 00000000, R9 is 00000000, R10 (s1) is 00000000, R11 (fp) is 00000000, R12 (ip) is 00000000, R13 (sp) is 00011400, R14 (lr) is 00000000, and R15 (pc) is 0000105e. Below the registers, the CPSR Register is shown with fields: Negative(N):0, Zero(Z):0, Carry(C):1, Overflow(V):0, IRQ Disable:1, FIQ Disable:1, Thumb(T):0, and CPU Mode: System.

**CodeView:** Displays assembly code for a program named P6.o. The code includes data sections for 'odd' and 'even' labels, a text section with instructions like LDR, MOV, and SWI, and a main execution loop with labels START, CONTINUE, and EVEN. The code ends with a B START instruction.

**OutputView:** Shows the console output of the program. The output text is: "Loading assembly language file C:\Nikhil\schoo1-work\Programs\4th Sem\MPCA\Week 4\P6.s", "Execution starting ...", "ODD PARITY", "Execution ending, Instruction Count:242 Elapsed Time:00:00:00.391364", and "Instructions per second:618".

RegistersView

General PurposeFloating Point

HexadecimalUnsigned DecimalSigned Decimal

R0:0000107b

R1:00000000

R2:00000002

R3:00000000

R4:00000000

R5:00000000

R6:00000000

R7:00000000

R8:00000000

R9:00000000

R10(sp):00000000

R11(fp):00000000

R12(lp):00000000

R13(sp):00011400

R14(lr):00000000

R15(pc):00001040

CPSR Register

Negative(N):0

Zero(Z):1

Carry(C):1

Overflow(V):0

IRQ Disable:1

FIQ Disable:1

Thumb(T):0

CPU Mode:System

0x600000df

CodeView

P6.o

.data

a: .word 20

odd: .asciz "ODD PARITY"

:49524150

:005954

even: .asciz "EVEN PARITY"

:52415020

:00595249

.text

LDR R0,=a

LDR R0,[R0]

MOV R1,#0

MOV R2,#0

MOV R3,#0

MOV R4,#0

START:

cmp R3,#32

BEQ END

TST R0,#1

BEQ CONTINUE

ADD R2,R2,#1

CONTINUE:

MOV R0,R0,LSR#1

ADD R3,R3,#1

B START

EVEN:

LDR R0,=even

SWI 0x02

SWI 0x011

OutputView

WatchView

MemoryView0

Console

Loading assembly language file C:\Nikhil\school-work\Programs\4th Sem\MPCA\Week 4\P6.s

Execution starting ...

www.dadguy

### **Disclaimer:**

- The programs and output submitted is duly written, verified and executed by me. I have not copied from any of my peers nor from the external resource such as internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Signature: Nikhil Girish

Name: Nikhil Girish

SRN: PES2UG21CS334

Section: 4F

Date: 13/02/23